```
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <stdlib.h>
#include <sys/types.h>
#include <netinet/in.h>
#define BUF_SIZE 256

int main(int argc, char *argv[])
{
    struct sockaddr_in server, client;
    char str[50], msg[20];
    if (argc != 2)
        printf("Input format not correct");
    int sockfd = socket(AF_INET, SOCK_DGRAM, 0);
    if (sockfd == -1)
        printf("Error in socket();");
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = INADDR_ANY;
    server.sin_port = htons(atoi(argv[1]));
    client.sin_family = AF_INET;
    client.sin_addr.s_addr = INADDR_ANY;
    client.sin_port = htons(atoi(argv[1]));
    if (bind(sockfd, (struct sockaddr *)&server, sizeof(server)) < 0)
        printf("Error in bind()! \n");
    socklen_t client_len = sizeof(client);
    printf("server waiting......");
    sleep(3);
    if (recvfrom(sockfd, str, 100, 0, (struct sockaddr *)&client, &client_len) <
0)
        printf("Error in recvfrom()!");
    printf("\nGot message from client:%s", str);
    printf("\nSending greeting message to client");
    strcpy(str, "220 127.0.0.1");
    sleep(10);
    if (sendto(sockfd, str, sizeof(str), 0, (struct sockaddr *)&client,
sizeof(client)) < 0)
        printf("Error in send");
    sleep(3);
    if ((recvfrom(sockfd, str, sizeof(str), 0, (struct sockaddr *)&client,
&client_len)), 0)
        printf("Error in recv");
    if (strncmp(str, "HELO", 4))
        printf("\n'HELO' expected from client....");
    printf("\n%s", str);
    printf("\nSending response...");
    strcpy(str, "250 ok");
    if (sendto(sockfd, str, sizeof(str), 0, (struct sockaddr *)&client,
sizeof(client)) < 0)
        printf("Error in send");
    sleep(3);
    if ((recvfrom(sockfd, str, sizeof(str), 0, (struct sockaddr *)&client,
&client_len)) < 0)
        printf("Error in recv");
    if (strncmp(str, "MAIL FROM", 9))
        printf("MAIL FROM expected from client...");
    printf("\n%s", str);
    printf("\nSending response....");
    strcpy(str, "250 ok");
    if (sendto(sockfd, str, sizeof(str), 0, (struct sockaddr *)&client,
sizeof(client)) < 0)
        printf("Error in send");
    sleep(3);
```

```
    if ((recvfrom(sockfd, str, sizeof(str), 0, (struct sockaddr *)&client,
&client_len)) < 0)
        printf("Error in recv");
    if (strncmp(str, "RCPT TO", 7))
        printf("\nRCPT TO expected from client....");
    printf("\n%s", str);
    printf("\nSending response....");
    strcpy(str, "250 ok");
    if (sendto(sockfd, str, sizeof(str), 0, (struct sockaddr *)&client,
sizeof(client)) < 0)
        printf("Error in send");
    sleep(3);
    if ((recvfrom(sockfd, str, sizeof(str), 0, (struct sockaddr *)&client,
&client_len)) < 0)
        printf("Error in recv");
    if (strncmp(str, "DATA", 4))
        printf("\nDATA expected from client....");
    printf("\n%s", str);
    printf("\nSending response....");
    strcpy(str, "354 Go ahead");
    if (sendto(sockfd, str, sizeof(str), 0, (struct sockaddr *)&client,
sizeof(client)) < 0)
        printf("Error in send");
    if ((recvfrom(sockfd, msg, sizeof(str), 0, (struct sockaddr *)&client,
&client_len)) < 0)
        printf("Error in recv");
    printf("mail body received");
    printf("\n%s", msg);
    if ((recvfrom(sockfd, str, sizeof(str), 0, (struct sockaddr *)&client,
&client_len)) < 0)
        printf("Error in recv");
    if (strncmp(str, "QUIT", 4))
        printf("quit expected from client....");
    printf("\nSending quit...");
    strcpy(str, "221 OK");
    if (sendto(sockfd, str, sizeof(str), 0, (struct sockaddr *)&client,
sizeof(client)) < 0)
        printf("Error in send");
    close(sockfd);
    return 0;
}
```